

Sabre

**SABRE
APIs**



Developer Administration Guide

**Market Fare View Crowd-Sourced
Batch Feed**

Itinerary Batch XML Schema Version 1.1.0

January 2021

© 2012-2021, Sabre Inc. All rights reserved.

This documentation is the confidential and proprietary intellectual property of Sabre Inc. Any unauthorized use, reproduction, preparation of derivative works, performance, or display of this document, or software represented by this document, without the express written permission of Sabre Inc. is strictly prohibited.

Sabre Travel Network and the Sabre Travel Network logo design are trademarks and/or service marks of an affiliate of Sabre Inc. All other trademarks, service marks, and trade names are owned by their respective companies.

Table of Contents

1 Getting Started

1.1 Introduction.....	1-1
1.2 About This Guide.....	1-1
1.3 Benefits	1-1
1.4 Customer Identification.....	1-1
1.5 Scope of Data.....	1-2
1.5.1 Data Aggregation.....	1-2
1.5.2 Global Product Version.....	1-2
1.5.3 Regional Product Version.....	1-3
1.6 Data Transfer	1-3

2 XML Structure

2.1 Introduction.....	2-1
2.2 Batch XML Format.....	2-1
2.3 Batch XML Field Descriptions	2-2

3 Set-Up Process

3.1 Introduction.....	3-1
3.2 Using the Set-Up Process	3-1

• • •

Getting Started

1.1 Introduction

The objective of Market Fare View Crowd-Sourced Batch (MFV) is to provide the lowest fares for each cabin class (economy, business, first) based on all fares that were shopped by Sabre customers during the latest hour.

The product can provide data aggregated on a global or regional basis.

The data is provided in a simple XML message format and made available in hourly batch feeds via secure FTP.

1.2 About This Guide

The purpose of this document is to describe how Market Fare View Crowd-Sourced Batch works and provide details on how to consume the feed. This document also provides details on the batch message structure. Each chapter is as follows:

- Chapter 1, “Getting Started” explains the purpose of the document, benefits, and details critical information.
- Chapter 2, “XML Structure” contains details about the message structure.
- Chapter 3, “Set-Up Process” contains the set-up process.

1.3 Benefits

Market Fare View Crowd-Sourced Batch Feed provides users with valuable insights into the cheapest fares selected based on all travel searches that were done by customers in the *Sabre* marketplace during the last hour. It aggregates significant amounts of organic (crowd-sourced) shopping data to make the lead fares for each cabin available via a standard, easy-to-use format. The data is provided for the customer, which prevents the customer from having to make numerous calls to Sabre.

Market Fare View Crowd-Sourced Batch data is filtered out and sorted by Sabre to deliver expected capabilities and to ensure that no legal and/or contractual bindings with other parties are violated.

1.4 Customer Identification

This service can be used by:

- *Sabre Dev Studio* subscribers.
- *Sabre Travel Network*® customers.
- Third-party customers.

1.5 Scope of Data

Market Fare View Crowd-Sourced Batch includes lead fares obtained from request and response pairs generated by shopping backend systems. No currency conversion is done or any other data transformation. Shopping data is collected during hourly interval, aggregated, filtered out, and sorted by Market Fare View according to the parameters specified in this guide and distributed to consumers.

1.5.1 Data Aggregation

The aggregation of data is done based on the unique combination of the following criteria:

- Market (origin and destination pair).
- Departure and return dates.
- Point of sale country.
- Currency.
- Cabin class (Economy, Business, First).
- Number of connections (non-stop vs connecting flights).

If a given combination of these criteria is shopped multiple times, the cheapest option within a one-hour batch period is forwarded.

1.5.2 Global Product Version

The global version of the product:

- Includes organic shopping data.
- Includes data from the following Point of Sale countries: US, CA, AD, AT, BE, CY, DE, ES, FI, FR, GR, IE, IT, LU, MC, MT, NL, PT, SI, SK, SM, AU, CN, ID, IN, JP, KR, MY, NZ, PH, SG, TH, TW, VN, AR, BR, CL, CO, MX, PE, VE, BG, CH, CZ, DK, EE, GB, HU, ME, NO, PL, RO, RU, SB, SE, TR, UA, XU, EG, IL, ZA.
- Includes round trips.
- Includes published fares only.
- Includes single passenger, adult fares.
- Includes Economy, Business, and First cabin classes.
- Excludes Southwest, easyJet, and Ryanair airline content.

1.5.3 Regional Product Version

The regional product version includes the same scope of data as the global version, but further filtered out to include data from certain POS regions only. The following regions are available:

- North America: US and CA.
- Euro currency zone: AD, AT, BE, CY, DE, ES, FI, FR, GR, IE, IT, LU, MC, MT, NL, PT, SI, SK, and SM.
- European countries outside of the Euro currency zone: BG, CH, CZ, DK, EE, GB, HU, ME, NO, PL, RO, RU, SB, SE, TR, UA, and XU.
- Middle East and Africa: EG, IL, and ZA.
- Latin America: AR, BR, CL, CO, MX, PE, and VE.
- Asia Pacific: AU, CN, ID, IN, JP, KR, MY, NZ, PH, SG, TH, TW, and VN.

1.6 Data Transfer

Data can be pushed/pulled over a file transfer protocol (sFTP).

• • •

2.1 Introduction

Market Fare View delivers *Sabre* Shopping Data in Batch XML format. This chapter includes the following:

- Batch XML format (see this page).
- Batch XML field descriptions (see “Batch XML Field Descriptions,” page 2-2).

2.2 Batch XML Format

```
<ItineraryBatch ShopDateTimeStart="2015-09-23T06:21:32.393-05:00"
ShopDateTimeEnd="2015-09-23T06:22:32.393-05:00">
  <AirItinerary ShopTime="2015-09-23T06:21:32.393-05:00"
ShopCount="10">
    <RequestInfo POSCountry="US">
      <Passenger Code="ADT" Number="1"/>
      <Passenger Code="JCB" Number="1"/>
      <Segment Origin="PVG" Destination="OSA"
DepartureDateTime="2015-09-25T00:00:00.000-05:00" Cabin="Y"/>
      <Segment Origin="OSA" Destination="PVG"
DepartureDateTime="2015-10-03T00:00:00.000-05:00" Cabin="Y"/>
    </RequestInfo>
    <ItineraryLeg>
      <FlightSegment Origin="PVG" Destination="KIX"
CarrierCode="K0" FlightNumber="163"
DepartureDateTime="2015-09-25T17:30:00.000+08:00"
ArrivalDateTime="2015-09-25T20:30:00.000+09:00"
Stops="0" BookingCode="G" CabinClass="Y"
SeatsRemaining="9" MarriedFlight="false"/>
    </ItineraryLeg>
    <ItineraryLeg>
      <FlightSegment Origin="KIX" Destination="PVG"
CarrierCode="K0" FlightNumber="164"
DepartureDateTime="2015-10-03T09:05:00.000+09:00"
ArrivalDateTime="2015-10-03T10:30:00.000+08:00"
Stops="0" BookingCode="U" CabinClass="Y"
SeatsRemaining="9" MarriedFlight="false"/>
    </ItineraryLeg>
    <Fare PassengerCode="ADT" PassengerNumber="1"
BaseAmount="378.00" TaxAmount="139.60" Currency="USD"
Private="false" Refundable="true" ETicketable="true">
      <FareSegment Start="PVG" End="KIX"
FareBasisCode="GR1MCC"/>
      <FareSegment Start="KIX" End="PVG"
FareBasisCode="UAB42CC"/>
      <Tax Amount="4.00" Code="YQI" Currency="USD"/>
    </Fare>
  </AirItinerary>
</ItineraryBatch>
```

```

        <Tax Amount="4.00" Code="YQI" Currency="USD"/>
        <Tax Amount="46.00" Code="YRF" Currency="USD"/>
        <Tax Amount="46.00" Code="YRF" Currency="USD"/>
        <Tax Amount="14.80" Code="CN1" Currency="USD"/>
        <Tax Amount="22.30" Code="SW" Currency="USD"/>
        <Tax Amount="2.50" Code="OI" Currency="USD"/>
    </Fare>
</AirItinerary>
<AirItinerary ShopTime="2015-09-23T06:21:35.425-05:00"
ShopCount="2">
    <RequestInfo POSCountry="RU">
        <Passenger Code="ADT" Number="1"/>
        <Segment Origin="LED" Destination="AER"
DepartureDateTime="2015-10-10T06:00:00.000-05:00" Cabin="Y"/>
        <Segment Origin="AER" Destination="LED"
DepartureDateTime="2015-10-19T06:00:00.000-05:00" Cabin="Y"/>
    </RequestInfo>
    <ItineraryLeg>
        <FlightSegment Origin="LED" Destination="AER"
CarrierCode="XX" FlightNumber="6563"
        DepartureDateTime="2015-10-10T14:40:00.000+03:00"
ArrivalDateTime="2015-10-10T17:55:00.000+03:00"
        Stops="0" BookingCode="E" CabinClass="Y"
SeatsRemaining="4" MarriedFlight="false"/>
    </ItineraryLeg>
    <ItineraryLeg>
        <FlightSegment Origin="AER" Destination="LED"
CarrierCode="XX" FlightNumber="6564"
        DepartureDateTime="2015-10-19T18:55:00.000+03:00"
ArrivalDateTime="2015-10-19T22:05:00.000+03:00"
        Stops="0" BookingCode="N" CabinClass="Y"
SeatsRemaining="9" MarriedFlight="false"/>
    </ItineraryLeg>
    <Fare PassengerCode="ADT" PassengerNumber="1"
BaseAmount="7500" TaxAmount="3464" Currency="RUB"
    Private="false" Refundable="false" ETicketable="true">
        <FareSegment Start="LED" End="AER"
FareBasisCode="EPXRF"/>
        <FareSegment Start="AER" End="LED"
FareBasisCode="NPXRF"/>
        <Tax Amount="1500" Code="YQF" Currency="RUB"/>
        <Tax Amount="1500" Code="YQF" Currency="RUB"/>
        <Tax Amount="232" Code="YRI" Currency="RUB"/>
        <Tax Amount="232" Code="YRI" Currency="RUB"/>
    </Fare>
</AirItinerary>
</ItineraryBatch>

```

2.3 Batch XML Field Descriptions

The following table contains descriptions in the XML message:

Field Name		Description	
Itinerary/ItineraryBatch		The ItineraryBatch message represents a batch of messages that are bundled together for more efficient network processing.	
	ShopDateTimeEnd	Date/time format is yyyy-mm-ddThh-mm-ss-msmsms followed by -/+hh-mm" to reflect time zones, for example 2012-09-04T14:31:00.000+02:00.	
	ShopDateTimeStart	Date/time format is yyyy-mm-ddThh-mm-ssmsmsms followed by -/+hh-mm" to reflect time zones, for example 2012-09-04T14:31:00.000+02:00.	
Itinerary/ItineraryBatch/Airltinerary		Ordered list of itineraries included in this Itinerary Batch.	
	ShopTime	Date and time of first shopping within the hour Date/time format is yyyy-mm-ddThh-mm-ss-msmsms followed by -/+hh-mm" to reflect time zones, for example 2012-09-04T14:31:00.000+02:00.	
	ShopCount	Number of occurrences a market/dates/ point of sale country/Currency/Cabin/ Number of connections was shopped during a given hour range.	
Itinerary/ItineraryBatch/Airltinerary/RequestInfo		Request information.	
		PosCountry	Point of sale country.
	Passenger		Information about passenger and requested market and date.
		Code	Passenger type code (for example, ADT for Adult).
		Number	Number of passengers requested.
	Segment		Ordered list of segments included in the request.
		Origin	Origin (departure) IATA airport code for the entire fare. The first airport of the first segment of the fare. In the case of a return fare, it also represents the last airport of the last segment.
		Destination	Destination (arrival) IATA airport code for the entire fare. This represents the destination of a one-way or return fare; leave blank in the case of multi-destination fares.
		DepartureDateTime	Departure date/time of this flight segment. Date/time format is yyyy-mm-ddThh-mm-ss-msmsms followed by -/+hh-mm" to reflect time zones (for example, 2012-09-04T14:31:00.000+02:00).

Field Name			Description
		Cabin	Cabin for this flight segment.
Itinerary/ItineraryBatch/AirItinerary/ItineraryLeg			Ordered list of legs included in this itinerary.
	Flight Segment		Ordered list of flight segments included in this leg, priced together under one or many fare codes and under single price.
		Origin	Origin (departure) of the flight segment.
		Destination	Destination (arrival) IATA airport code for the entire fare. This represents the destination of a one-way or return fare; leave blank in the case of multi-destination fares.
		CarrierCode	Two-character IATA airline code for the carrier of this flight segment.
		FlightNumber	Flight number for this flight segment. Usually between 1 and 4 digits.
		DepartureDateTime	Departure date/time of this flight segment. Date/time format is "yyyy-mm-ddThh-mm-ssmsmsms" followed by "-/+hh-mm" to reflect time zones (for example, 2012-09-04T14:31:00.000+02:00).
		ArrivalDateTime	Arrival date/time of this flight segment. Date/time format is "yyyy-mm-ddThh-mm-ssmsmsms" followed by "-/+hh-mm" to reflect time zones (for example, 2012-09-04T14:31:00.000+02:00).
		Stops	Number of stops on this flight segment. Usually 0 unless the airplane makes a stop to refuel. A change in aircraft or flight number indicates a new segment rather than a stop.
		CabinClass	Cabin class, also known as the RBD, or IATA Class Code.
		SeatsRemaining	Number of available seats remaining for sale at this fare price.
		BookingCode	Class of service.
		MarriedFlight	Indicates whether the current flight segment is married to the previous flight segment, or it means that the current flight segment is part of married segments.
Itinerary/ItineraryBatch/AirItinerary/Fare			Fare information.
		PassengerCode	Passenger type code (for example, ADT for Adult).
		BaseAmount	Base fare price (not including TaxAmount).

Field Name		Description	
		TaxAmount	Additional tax and surcharges that must be added on top the base fare price.
		Currency	International currency code for the BaseAmount and TaxAmount values.
		Private	Indicator for whether this fare is private. Default is false.
		Refundable	Indicator for whether this fare is refundable. Default is false.
		ETicketable	Indicates that the fare within the itinerary is e-ticketable (T) or indicates that the fare within the itinerary is not e-ticketable (F).
	FareSegment		Ordered list of fare segments included in this fare
		Start	Origin (departure) IATA airport code for this flight segment.
		End	Destination (arrival) IATA airport code for this flight segment.
		FareBasisCode	Fare code, also known as the FBC or Fare Basis Code (for example, H14ESNR).
		PassengerCode	Passenger type code (for example, ADT for Adult).
	Tax		List of taxes included in this fare.
		Amount	Tax amount.
		Code	Tax code.
		Currency	Currency.

• • •

Set-Up Process

3.1 Introduction

This chapter includes the set-up process for customers.

3.2 Using the Set-Up Process

1. The customer provides the connectivity details in one of the EFG data collectors below (use PUSH if customer wants the data pushed to their server; use PULL if customer wants to pull the data from Sabre server).



EFG_Data_Collector
_PUSH.docx



EFG_Data_Collector
_PULL.docx

2. For PULL mode, Sabre supplies sFTP accounts for non-PROD and PROD environments.
3. For PULL mode, the customer performs validation testing on both environments.
4. For PUSH mode, Sabre supplies public keys if this authentication method is selected and performs validation testing on both environments.

• • •